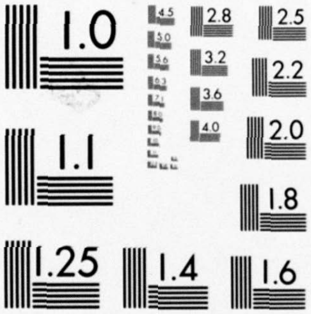
AD-A082 388   CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 9/3
THE AREA-TIME COMPLEXITY OF BINARY MULTIPLICATION,(U)
JUL 79 R P BRENT, H T KUNG   N00014-76-C-0370

UNCLASSIFIED CMU-CS-79-136   NL

| OF |
AD-
A082388

END
DATE
FILMED
4-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The Area-Time Complexity of Binary Multiplication

R.P. Brent H.T. Kung

July 1979

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

14 CMU-CS-79-136

6

The Area-Time Complexity of Binary Multiplication

10

R.P. Brent
Department of Computer Science
Australian National University
Canberra, A.C.T. 2600
Australia

H.T. Kung
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213
U.S.A.

11 July[redacted]79

12 27

10

(Published simultaneously as a technical report, TR-CS-79-05, Department of Computer Science, Australian National University.)

15 NØØØ14-76-C-Ø37Ø

This research was supported in part by the National Science Foundation under Grant MCS 78-236-76 and the Office of Naval Research under Contract N00014-76-C-0370. Most of this work was carried out at the Australian National University while H.T. Kung was visiting there as a Visiting Fellow during May 1979.

JOB

403081

Abstract

We consider the problem of performing multiplication of n-bit binary numbers on a chip. Let A denote the chip area, and T the time required to perform multiplication. Using a model of computation which is a realistic approximation to current and anticipated VLSI technology, we show that

alpha is an element [0,1>   sub∅   to the 2alpha power

$$(A/A_0)(T/T_0)^{\boxed{2\alpha}} \geq n^{\boxed{1+\alpha}}$$

to the (1+alpha) power

sub∅   > or =

for all $\alpha \in [0,1]$, where A_0 and T_0 are positive constants which depend on the technology but are independent of n. The exponent $1+\alpha$ (alpha) is the best possible. A consquence is that binary multiplication is "harder" than binary addition if $AT^{\boxed{2\alpha}}$ is used as a complexity measure for any $\alpha \geq 0$.

to the 2alpha power

alpha   > or =

Key Words and Phrases

Area-time complexity, binary multiplication, chip layout, circuit design, combinational logic, computational complexity, lower bounds, multiplication, VLSI.

CR Categories

5.25, 6.1, 6.32.

1. Introduction

We are interested in the design of multipliers suitable for implementation in VLSI architecture. The multiplication problem has been considered by several authors, e.g. Garner [76], Kuck [78], Ofman [62], Wallace [64], and Winograd [67]. Much attention has been paid to the tradeoff between time and the number of gates, but until recently little attention has been paid to the problem of connecting the gates in an economical and regular way to minimize chip area and design costs.

In this paper we give a lower bound on the area-time product for multiplication circuits, assuming a model of computation which is intended to approximate current and anticipated VLSI technology. Details of the model are given in Section 2.

The lower bound on AT, where A is the chip area and T the time to perform n-bit binary multiplication on the chip, is the special case $\alpha = \frac{1}{2}$ of a more general lower bound

$$\left(\frac{A}{A_0}\right)\left(\frac{T}{T_0}\right)^{2\alpha} \geq n^{1+\alpha} \tag{1.1}$$

which is valid for all $\alpha \in [0,1]$. We establish the lower bound for the extreme cases $\alpha = 1$ and $\alpha = 0$ in Sections 3 and 4 respectively, and deduce the general result in Section 5.

In this paper we are concerned with lower bounds, and do not give any practical designs for multiplication circuits. In Section 6 we sketch an (impractical) design which has $AT^{2\alpha} = O(n^{1+\alpha+\epsilon})$ for any $\epsilon > 0$. Thus, the exponent $1 + \alpha$ in (1.1) is sharp, although we do not know any practical design for which the lower bound is approached.

In Brent and Kung [79] we give an upper bound on A and T for the problem of addition of n-bit binary numbers. From this result it follows that binary multiplication is "harder" than binary addition if the complexity measure is $AT^{2\alpha}$ (for any $\alpha \geq 0$).

2. The Computational Model

Our model is intended to be general, but at the same time realistic enough to apply to current VLSI technology such as MOS. We assume the existence of circuit elements or "gates" which compute a logical function of two inputs in constant time and occupy at least a constant minimum area. Gates are connected by wires which have constant minimum width (or, equivalently, must be separated by at least some minimal spacing). Our measure of the cost of a design is the area rather than the number of gates required. This is an important difference between our model and earlier models of Winograd [67], Brent [70] and others.

Assumptions

In Sections 3 to 5 we need various subsets of the following assumptions A1 to A8. Comments and justification are given in parentheses following the statement of each assumption. Our notation is summarized in the Appendix.

A1. The computation is performed in a convex planar region R of area A. {Because of heat-dissipation and packing requirements, a two-dimensional planar model is reasonable. If R is not convex we may take its convex hull. R may be a whole chip or part of a chip.}

A2. Wires have minimal width $\lambda > 0$. {λ is assumed constant, but in applications of our results it will of course depend on the technology: see Mead and Conway [79].} We also assume R has width at least λ.

A3. At most $\nu \geq 2$ wires can overlap at any point in R. {Otherwise the area could be reduced by "folding". Since $\nu \geq 2$, the graph of wires (edges) and gates (nodes) need not be planar in a graph-theoretic sense.}

A4. A bit requires minimal time $\tau > 0$ to propagate along a wire. The time for one gate computation and an arbitrary fan-out of the result is included in τ. {Since dimensions are limited by the minimal wire-width λ and minimal gate area, a minimal propagation time is reasonable. We do not need to assume that the propagation time increases with the length of the wire. This is fortunate, for with current technology propagation times are limited by wire capacitances rather than the velocity of light. A longer wire will generally have a larger capacitance, and thus require a larger driver to maintain constant propagation time, but the driver area need not exceed a fixed percentage of the wire area, so can be ignored if λ is increased slightly; see Mead and Conway [79]. Although it would be reasonable to assume bounded fanout, we do not need this assumption for proving lower bounds. When proving upper bounds, we do assume bounded fanout.}

A5. I/O ports each have area at least $\rho \geq \lambda^2$. {If R is a complete chip, ρ will be large compared to λ^2. If R is only part of a chip and I/O is to other regions on the chip, ρ could be of order λ^2.

A6. Storage for one bit of information takes area at least $\beta > 0$. {β will typically be larger than λ^2, but we do not need to assume this.}

A7. Each input bit is available only once. {There is no free memory outside R. If the same input bit is required at different times, it must be stored within R, taking area at least β (see A6).}

A8. The times and locations at which input and output bits are available are fixed and independent of the values of the input bits. {This is necessary if designs are to be modular.}

3. A Lower Bound on AT^2

With the model of Section 2, we have the following lower bound on AT^2 for any multiplier circuit.

Theorem 3.1

Under assumptions A1 to A5,

$$AT^2 \geqslant K_1 n^2, \tag{3.1}$$

where

$$K_1 = \frac{4}{\pi}\left(\frac{\lambda\tau}{(9+4\sqrt{5})\nu}\right)^2 . \tag{3.2}$$

Before proving Theorem 3.1 we need three Lemmas.

Lemma 3.1

For any convex planar figure with area A, perimeter L, diameter D, and chord of length C perpendicular to D,

$$A \geqslant \frac{CL}{2\pi} \tag{3.3}$$

and

$$L \geqslant 2\sqrt{\pi A} \qquad (3.4)$$

Proof

The results follow from the well-known inequalities $A \geqslant CD/2$, $\pi D \geqslant L$ and $L^2 \geqslant 4\pi A$. For a proof (and a definition of "diameter" etc) see Yaglom and Boltyanskii [61]. ■

Lemma 3.2

$$\min_{0 \leq r < 1} \max(16r, (1-r)^2) = \frac{16}{9+4\sqrt{5}} .$$

Proof

It is easy to verify that the minimum occurs when $16r = (1-r)^2$, and the only root of this equation in $(0,1)$ is $r = 1/(9+4\sqrt{5})$. ■

Lemma 3.3

Suppose that less than n outputs share any one output port. Then, under assumptions A1 to A4,

$$AT \geqslant K_2 Ln$$

where

$$K_2 = \frac{\lambda\tau}{(9+4\sqrt{5})\pi\nu} . \qquad (3.5)$$

Proof

Let $S = \{p_{2n-1}, \ldots, p_n\}$ where, in binary notation, $p_{2n} \ldots p_1$ is the 2n-bit product of the n-bit numbers $a_n \ldots a_1$ and $b_n \ldots b_1$.

Let M be the maximum number of elements of S sharing any one output port. (By assumption, $1 \leqslant M < n$.) Let D be a diameter of R, and C a chord perpendicular to D, dividing S into two parts S_1 and S_2 such that the output ports for elements of S_1 lie on one side of C, and those for elements of S_2 lie on the other side of C. Since we do not use assumption A6, we can assume that output ports are shrunk to infinitesimal size and that (by an infinitesimal perturbation from the perpendicular to D) C does not intersect any output ports. By "sliding" the intersection of C and D along D, we can arrange that

$$(3.6) \qquad \left\lceil \frac{n-M}{2} \right\rceil \leqslant |S_i| \leqslant \left\lfloor \frac{n+M}{2} \right\rfloor \qquad \text{for } i = 1,2.$$

Consider multipliers $b = 2^j$ for $j=0,1,\ldots,n-1$. Multiplying $a = a_n \ldots a_1$ by 2^j gives $p_{i+j} = a_i$ for $i=1,\ldots,n$. Consider a fixed a_i with $\left\lfloor \frac{n+M}{2} \right\rfloor \leqslant i \leqslant n$. For $j=n-i,\ldots,n-1$, we have $n \leqslant i+j < 2n$, so $p_{i+j} \in S$. Let

$$S_3(i) = \begin{cases} S_1 & \text{if the input port for } a_i \text{ is on the same side} \\ & \text{of C as the output ports for elements of } S_1, \\ S_2 & \text{otherwise.} \end{cases}$$

As j ranges over $n-i,\ldots,n-1$, at most $\left\lfloor \frac{n+M}{2} \right\rfloor$ of the p_{i+j} lie in $S_3(i)$ (by (3.6)), so at least $i - \left\lfloor \frac{n+M}{2} \right\rfloor$ lie in $S-S_3(i)$. Thus, by definition of $S_3(i)$, a bit of information must cross C for each such p_{i+j}. Summing over $i = \left\lfloor \frac{n+M}{2} \right\rfloor,\ldots,n$, a total of at least $0+1+2+\ldots+\left(n - \left\lfloor \frac{n+M}{2} \right\rfloor\right) \geqslant \frac{(n-M)^2}{8}$ bits must cross C. Since there are only n possible values of j, there is some j for which at least $\frac{(n-M)^2}{8n}$ bits must cross C before the product of a and $b=2^j$ can be transmitted through the output ports.

By assumptions A2 and A3, at most $\nu C/\lambda$ wires cross C. Thus, by assumption A4,

$$\left(\frac{\nu C}{\lambda}\right)\left(\frac{T}{\tau}\right) \geq \frac{(n-M)^2}{8n} .$$

It follows from (3.3) that

$$AT \geq \frac{\lambda\tau L n(1-r)^2}{16\pi\nu} \tag{3.7}$$

where

$$r = M/n.$$

Since M outputs come through one output port, assumption A4 gives

$$T \geq M\tau. \tag{3.8}$$

Also, since $M < n$, at least one wire crosses C, and assumption A2 gives

$$C \geq \lambda. \tag{3.9}$$

By assumption A3, $\nu \geq 2$. Combining this with (3.3), (3.8) and (3.9) gives

$$AT \geq \frac{\lambda\tau L n r}{\pi\nu} . \tag{3.10}$$

The result now follows from (3.7), (3.10) and Lemma 3.2. ■

Lemma 3.3 is of interest in its own right. If at one time the chip inputs or outputs a total of b bits along its boundary, then $L \geq b\lambda$ and the lemma gives $AT \geq K_2\lambda bn$. Thus for any multiplication scheme that accepts, say, $n^{1/2}$ input bits simultaneously along the chip boundary, we know immediately that $AT \geq (K_2\lambda)n^{3/2}$ (cf. the multiplication scheme in Section 6).

Proof of Theorem 3.1

Let M be as in the proof of Lemma 3.3. If $M = n$, then n output bits share one output port, and assumption A4 gives $T \geq \tau n$. Since there is at least one output port, assumption A5 gives $A \geq \rho \geq \lambda^2$, so

$$AT^2 \geq (\lambda\tau n)^2 > K_1 n^2 \ . \tag{3.11}$$

If $M < n$ then Lemma 3.3 is applicable, and gives

$$AT \geq K_2 Ln$$

so, from (3.4),

$$AT \geq 2K_2(\pi A)^{1/2} n,$$

and thus

$$AT^2 \geq 4\pi K_2^2 n^2 \ . \tag{3.12}$$

The result follows from (3.5), (3.11) and (3.12). ∎

Theorem 3.1 (with a smaller constant for K_1) could have been established by a proof parallel to that used by Thompson [79] for the DFT problem. In fact, using his result that relates the area of a graph to its minimum bisection width, one can prove Theorem 3.1 without the convexity assumption in A1. Our proof, above, represents a new approach that incorporates geometric considerations in the lower bound proof. We feel that the extra convexity assumption we make is not restrictive, since most existing chips do have convex boundaries for packaging reasons. Furthermore, we note that the convexity assumption is needed for establishing results such as Lemma 3.3 that relates AT to the perimeter.

4. A Lower Bound on the Area A

In Section 3 we gave a lower bound on AT^2. Now, using different techniques, we give a lower bound on A.

Theorem 4.1

Under assumptions A5 to A8,

$$A \geq A_0 n , \tag{4.1}$$

where

$$A_0 = \frac{5\beta\rho}{6(\beta+\rho)} . \tag{4.2}$$

Let $P_N = \{ij \mid 0 \leq i < N, 0 \leq j < N\}$ be the set of all integers which can be written as a product of two factors, each less than N; and let $\mu(N) = |P_N|$ be the cardinality of P_N. For example, $P_4 = \{0,1,2,3,4,6,9\}$ and $\mu(4) = 7$. Before proving Theorem 4.1 we need lower bounds on $\mu(N)$ and a related function

$$\delta(n) = \lceil \lg \mu(2^n)+1-n \rceil /n. \tag{4.3}$$

Lemma 4.1

$$\mu(N) \geq \sigma(N),$$

where $\sigma(N) = \sum_{p\in P_{N-1}} p$ and P_{N-1} is the set of prime numbers p in the range $2 \leq p < N$.

Proof

The numbers pj are distinct if $2 \leq p < N$, p prime, and $1 \leq j \leq p$. Thus, the result follows from the definition of $\mu(N)$. ■

Lemma 4.2

$$\mu(N) \geq \frac{N^2}{2\ln(N)} \qquad \text{for all } N \geq 4.$$

Proof

Using a slight modification of Theorem 1 and equation (4.13) of Rosser and Schoenfeld [62], we can show that

$$\sigma(N) > \frac{N^2}{2\ln(N)} \qquad \text{for all } N \geq 348.$$

Thus, the result for $N \geq 348$ follows from Lemma 4.1. For $4 \leq N \leq 347$, the result may be verified by a straightforward computation. ■

Lemma 4.3

If $\delta(N)$ is defined by (4.3), then

$$\delta(n) \geq \frac{5}{6} \qquad \text{for all } n \geq 1.$$

Proof

From Lemma 4.2,

$$\delta(n) \geq \lceil n - \lg(n\ln 2)\rceil / n, \tag{4.4}$$

and it is easy to verify that the right side of (4.4) is at least 5/6 for all $n \geq 18$. (There is equality for $n = 18$ and $n = 24$.) For $1 \leq n \leq 17$, direct computation shows that $\delta(n) \geq 9/10$. ■

Table 4.1 gives $\mu(2^n), \mu(2^n)/\hat{\mu}(2^n)$, and $\delta(n)$ for $n=1,2,\ldots,17$, where

$$\hat{\mu}(N) = \frac{N^2}{0.71 + \lg\lg N} \tag{4.5}$$

is an empirical approximation to $\mu(N)$. For $5 \leq n \leq 17$, the approximation error is less than 1 percent. If this remained true for $n > 17$, it would follow that $\delta(n) \geq 9/10$, and the constant 5/6 in Lemma 4.3 and Theorem 4.1

could be increased. On the basis of the empirical evidence, we make the following conjectures.

Conjecture 4.1

$$\delta(n) \geqslant 9/10 \qquad \text{for all } n \geqslant 1.$$

Conjecture 4.2

$$\lim_{N\to\infty} \frac{\mu(N)\lg\lg N}{N^2} = 1.$$

n	$\mu(2^n)$	$\mu(2^n)/\hat{\mu}(2^n)$	$\delta(n)$
1	2	0.355000	1
2	7	0.748125	1
3	26	0.932329	1
4	90	0.952734	1
5	340	1.006695	1
6	1,238	0.995890	1
7	4,647	0.997629	1
8	17,578	0.995092	1
9	67,592	1.000412	1
10	259,768	0.998846	9/10
11	1,004,348	0.998392	10/11
12	3,902,357	0.999002	11/12
13	15,202,050	0.999089	12/13
14	59,410,557	0.999788	13/14
15	232,483,840	0.999637	14/15
16	911,689,012	0.999788	15/16
17	3,581,049,040	1.000005	16/17

Table 4.1 $\mu(2^n)$ and related functions for $n = 1(1)17$.

Proof of Theorem 4.1

If $n = 1$ there is at least one output port, so $A \geq \rho$, and the result holds. Hence, suppose that $n \geq 2$.

Consider the state of the computation just before the last input bit(s) are accepted. Let m be the number of input bits still to be accepted, so $1 \leq m \leq 2n$.

It is easy to show that there are some inputs a and b such that the output bits $p_{2n},\ldots,p_n$ are not determined by the $2n-m$ input bits already accepted. Thus, by assumption A8, at most $n-1$ bits $(p_{n-1},\ldots,p_1)$ have been output.

Suppose that s bits of information are stored in R. Then we must have by assumption A7

$$\mu(2^n) \leq 2^{m+(n-1)+s} ,$$

or the circuit could not produce all $\mu(2^n)$ possible outputs, and would fail for certain inputs. Thus

$$m+s \geq \lceil \lg \mu(2^n)+1-n \rceil = n\delta(n).$$

and, from Lemma 4.3,

$$m+s \geq 5n/6. \tag{4.6}$$

By assumption A6,

$$A \geq \beta s. \tag{4.7}$$

Since a port can accept only one bit at a time, the last m bits must be input through m different ports, so A5 gives

$$A \geq \rho m. \tag{4.8}$$

The result follows easily from (4.6), (4.7) and (4.8). ∎

5. A General Lower Bound Result

Theorems 3.1 and 4.1 are the extreme cases $\alpha = 1$ and $\alpha = 0$ of the following result.

Theorem 5.1

Under assumptions A1 to A8, for all $\alpha \in [0,1]$,

$$\left(\frac{A}{A_0}\right)\left(\frac{T}{T_0}\right)^{2\alpha} \geq n^{1+\alpha} . \tag{5.1}$$

Here A_0 is given by (4.2),

$$T_0 = (K_1/A_0)^{\frac{1}{2}} , \tag{5.2}$$

and K_1 is given by (3.2).

Proof

From Theorem 3.1,

$$(A/A_0)\ (T/T_0)^2 \geq n^2 ,$$

so

$$(A/A_0)^{\alpha}\ (T/T_0)^{2\alpha} \geq n^{2\alpha} . \tag{5.3}$$

From Theorem 4.1,

(5.4) $$(A/A_0)^{1-\alpha} \geq n^{1-\alpha} .$$

Multiplying (5.3) and (5.4) gives the result. ∎

The following Corollary of Theorem 5.1 seems worth stating separately, for AT is often used as a complexity measure (see, e.g., Mead and Rem [79]).

Corollary 5.1

Under assumptions A1 to A8,

$$AT \geq K_3 n^{3/2} ,$$

where

(5.5) $$K_3 = A_0 T_0 = (A_0 K_1)^{\frac{1}{2}} .$$

Remarks

1. The constants in Theorem 5.1 and Corollary 5.1 are

$$A_0 = 0.83h,$$

$$T_0 = \frac{0.068\lambda\tau}{\nu h^{\frac{1}{2}}} ,$$

and

$$K_3 = \frac{0.057\lambda\tau h^{\frac{1}{2}}}{\nu} ,$$

where

$$h = \beta\rho/(\beta+\rho) .$$

Note that $\frac{1}{2}\min(\beta,\rho) \leq h < \min(\beta,\rho)$.

2. By the method of Section 6, we can perform binary multiplication with $A = 0(n\lg^2 n)$, $T = 0(n^{½}\lg^2 n)$, and

$$AT^{2\alpha} = 0(n^{1+\alpha}\lg^{2+4\alpha} n). \tag{5.5}$$

Thus, the exponent $1+\alpha$ in Theorem 5.1 is the best possible.

3. By a straightforward method we can achieve $A = 0(n^2 \lg n)$, $T = 0(\lg n)$, so

$$AT^{2\alpha} = 0(n^2 \lg^{1+2\alpha} n).$$

Thus, Theorem 5.1 can not be extended for $\alpha > 1$.

4. In Brent and Kung [79] we show that, for n-bit binary addition,

$$AT^{2\alpha} = \begin{cases} 0(n^{2\alpha}) & \text{if } 0 \leq \alpha \leq ½ , \\ 0(n\lg^{1+2\alpha} n) & \text{if } \alpha > ½ . \end{cases}$$

Thus, binary addition is easier than binary multiplication, for all complexity measures $AT^{2\alpha}$, $\alpha \geq 0$. (This holds for $\alpha > 1$ because $AT^{2\alpha} > AT^2 \geq K_1 n^2$.)

6. An Upper Bound on $AT^{2\alpha}$

It is easy to design practical n-bit multipliers with area $A = 0(n)$ and time $T = 0(n)$, so

$$AT^{2\alpha} = 0(n^{1+2\alpha}) . \tag{6.1}$$

In this section we sketch the design of a multiplier with $A = 0(n\lg^2 n)$ and $T = 0(n^{½}\lg^2 n)$, giving

$$AT^{2\alpha} = 0(n^{1+\alpha} \lg^{2+4\alpha} n), \tag{6.2}$$

which is asymptotically better than (6.1). The design is not practical, but it is theoretically interesting because it shows that the exponent $1 + \alpha$ in Theorem 5.1 is sharp. We do not know if there is any practical design having $AT^{2\alpha} = o(n^{1+2\alpha})$. Straightforward implementations of "fast" serial algorithms, e.g. the Schönhage-Strassen algorithm (Schönhage and Strassen [71]), or the "3-2 reduction" algorithm (Ofman [62]) seem to require area at least order n^2.

In the remainder of this section we assume:

1. $n = k^2$ is a perfect square, and
2. $a_j = b_j = 0$ if $j > n/2$.

(If not, n may be increased sufficiently without affecting the asymptotic results.) Let p be the smallest prime of the form $nq+1$, $q \geqslant 1$, F_p the finite field of integers mod p. We assume that $\lg p = O(\lg n)$, which is certainly true in practice, as $q \leqslant 84$ for all $n \leqslant 10000$. (If $\lg n \neq O(\lg n)$ we replace F_p by the complex field and work to sufficient accuracy to get the required results c_j at Step 5 below, or use other methods described in Adleman *et al* [78], Borodin and Munro [75], and Aho *et al* [74].) Let u be an n-th root of unity in F_p, and $w = u^k$ (so w is a k-th root of unity). Note that in any circuit n is fixed, so we are not concerned with the complexity of finding p, u etc: they will be encoded into the circuit.

In Steps 1-5 below all arithmetic is done in F_p. In Steps 1-3 we compute the Fourier transform $\underline{\hat{a}}$ of $(a_1,\ldots,a_n)$ and $\underline{\hat{b}}$ of $(b_1,\ldots b_n)$ over F_p, i.e. $$\hat{a}_{j+1} = \sum_{i=0}^{n-1} a_{i+1} u^{ij}$$

for $j=0,\ldots,n-1$, etc.

In Step 4 we multiply the Fourier transforms. In Step 5 we take the inverse transform, and in Step 6 the final result is computed.

Step 1

Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{U}$, and $\mathcal{W}$ be k by k matrices with elements

$$\mathcal{A}_{ij} = a_{(i-1)k+j},$$

$$\mathcal{B}_{ij} = b_{(i-1)k+j},$$

$$\mathcal{U}_{ij} = u^{(i-1)(j-1)},$$

and $$\mathcal{W}_{ij} = w^{(i-1)(j-1)}.$$

Perform k by k matrix multiplications to compute

$$\mathcal{A}' = \mathcal{W}\mathcal{A} \quad \text{and} \quad \mathcal{B}' = \mathcal{W}\mathcal{B},$$

using the "hexagonal array" scheme of Kung and Leiserson [79]. All computations are performed in F_p, so each processing element of the hexagonal array needs to perform arithmetic operations in F_p. Operations in F_p require no more than area $O(\lg^2 p)$ and time $O(\lg^2 p)$. Thus, Step 1 can be done with area $O(n\lg^2 n)$ and time $O(n^{\frac{1}{2}}\lg^2 n)$.

Step 2

Compute $\mathcal{A}'' = \mathcal{A}' \circ \mathcal{U}$ and $\mathcal{B}'' = \mathcal{B}' \circ \mathcal{U}$, where o denotes componentwise multiplication.

Step 3

Compute $A''' = A''W$ and $B''' = B''W$ using the same method as for Step 1. It may be shown that A''' and B''' contain the Fourier transforms of $(a_1,\ldots,a_n)$ and $(b_1,\ldots,b_n)$, in fact

$$A'''_{ij} = \hat{a}_{(j-1)k+i}$$

and $$B'''_{ij} = \hat{b}_{(j-1)k+i} \qquad \text{for } 1 \leq i,j \leq k.$$

Step 4

Compute $C''' = A''' \circ B'''$.

Step 5

Compute $C = W^{-1}(U' \circ (C'''W^{-1}))$ as in Steps 1-3. Here $U'_{ij} = u^{-(i-1)(j-1)}$. C represents the inverse Fourier transform of C'''. If

$$C_{ij} = c_{(i-1)k+j}$$

then, by the convolution Theorem and our assumption 2 above,

$$c_j = a_1b_j + a_2b_{j-1} + \ldots + a_jb_1 \qquad \text{for } 1 \leq j \leq n.$$

Thus, $$\sum_{i=1}^{2n} p_i 2^{i-1} = \sum_{i=1}^{n} c_i 2^{i-1} ,$$

and the problem of computing $p_{2n},\ldots,p_1$ has been reduced to the problem of summing $O(\lg n)$ numbers of at most $2n$ bits (since the c_i have $O(\lg n)$ bits). Hence, the final step in the computation is:

Step 6

Compute $p_{2n},\ldots,p_1$ from the c_i.

This may be done by $O(\lg n)$ additions, each requiring area $O(\lg n)$ and time $O(\lg n)$: see Brent and Kung [79].

This completes our outline of the multiplier with area $A = O(n\lg^2 n)$, time $T = O(n^{1/2}\lg^2 n)$, and $AT^{2\alpha} = O(n^{1+\alpha}\lg^{2+4\alpha} n)$. The exponent $2+4\alpha$ of lgn can certainly be reduced, but we do not know what its minimal value is.

7. Some Open Problems

Our results suggest several interesting problems:

1. Can the constants A_0 and T_0 be increased?
2. How far can the gap $O(\lg^{2+4\alpha} n)$ between the upper and lower bounds be reduced?
3. Is there a practical design with $AT^{2\alpha} = O(n^{1+\alpha+\varepsilon})$, for all $\varepsilon > 0$?
4. Can any of our assumptions A1 to A8 be relaxed?
5. Can the restriction to binary representation be removed?
6. For binary division it is easy to deduce a lower bound of the same form as (5.1), using the method of Brent [76]; and an upper bound $AT^{2\alpha} = O(n^{1+\alpha}\lg^{2+6\alpha} n)$, using Newton's method. Thompson [79] has proved a lower bound like (5.1) for computation of the discrete Fourier transform, using a model similar (though not identical) to ours. Can similar upper and/or lower bounds be proved for other computations?

References

Adleman, L., Booth, K.S., Preparata, F.P. and Ruzzo, W.L. [1978], "Improved time and space bounds for Boolean matrix multiplication", *Acta Informatica* 11, 61-75.

Aho, A.V., Hopcroft, J.E. and Ullman, J.D. [1974], *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Menlo Park, Ca.

Borodin, A. and Munro, I. [1975], *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York.

Brent, R.P. [1970], "On the addition of binary numbers", *IEEE Trans. Comp.* C-19, 758-579.

Brent, R.P. [1976], "The complexity of multiple-precision arithmetic", in *The Complexity of Computational Problem Solving* (edited by R.S. Anderssen and R.P. Brent), Univ. of Queensland Press, Brisbane, 126-165.

Brent, R.P. and Kung, H.T. [1979], "A regular layout for parallel adders", Tech. Report, Department of Computer Science, Carnegie-Mellon Univ., Pittsburgh, June 1979. Submitted to *IEEE Trans. Comp.*

Garner, H.L. [1976], "A survey of some recent contributions to computer arithmetic", *IEEE Trans. Comp.* C-25, 1277-1282.

Kuck, D.J. [1978], *The Structure of Computers and Computations*, Vol. 1, John Wiley and Sons, New York, 1978.

Kung, H.T. and Leiserson, C.E. [1979], "Systolic arrays for (VLSI)", In Sparse Matrix Proceedings 1978, (edited by I.S. Duff and G.W. Stewart), SIAM, 256-282. A slightly different version appears in Mead and Conway [79].

Mead, C.A. and Conway, L.A. [1979], *Introduction to VLSI Systems*, Addison-Wesley, Menlo Park, Ca.

Mead, C.A. and Rem, M. [1979], "Cost and Performance of VLSI Computing Structures", IEEE Journal of Solid State Circuits, SC-14(2), 455-462.

Ofman, Yu. [1962], "On the algorithmic complexity of discrete functions", *Dokl. Akad. Nauk SSSR* 145, 48-51. (In Russian.)

Rosser, J.B. and Schoenfeld, L. [1962], "Approximate formulas for some functions of prime numbers", *Illinois J. Math.* 6, 64-94.

Schönhage, A. and Strassen, V. [1971], "Schnelle Multiplikation grosser Zahlen", *Computing* 7, 281-292.

Thompson, C.D. [1979], "Area-time complexity for VLSI", *Proc. 11th Annual ACM Symposium on the Theory of Computing*, ACM, New York, 81-88.

Wallace, C.S. [1964], "A suggestion for a fast multiplier", *IEEE Trans. Elec. Comp.* EC-13, 14-17.

Winograd, S. [1967], "On the time required to perform multiplication", *J. ACM* 14, 793-802.

Yaglom, I.M. and Boltyanskii, V.G. [1961], *Convex Figures*, Holt, Rinehart and Winston, New York. (English translation by P.J. Kelly and L.F. Walton.)

Appendix : Summary of Notation

a : input to multiplier, $0 \leq a < 2^n$, $a = a_n \ldots a_1$ in binary notation.

a_i : i-th least significant bit of a, $1 \leq i \leq n$.

$\hat{a}_i$: $(\hat{a}_1, \ldots, \hat{a}_n)$ is Fourier transform of $(a_1, \ldots, a_n)$ over F_p.

A : area of region R. See assumption A1.

A_0 : $5h/6$.

$\mathcal{A}$: k by k matrix (A_{ij}). See Section 6. Similarly for $\mathcal{A}'$, $\mathcal{A}''$ and $\mathcal{A}'''$.

b : input to multiplier, $0 \leq b < 2^n$, $b = b_n \ldots b_1$.

b_i : i-th least significant bit of b, $1 \leq i \leq n$.

$\hat{b}_i$: $(\hat{b}_1, \ldots, \hat{b}_n)$ is Fourier transform of $(b_1, \ldots, b_n)$ over F_p.

$\mathcal{B}$: k by k matrix. See Section 6. Similarly for $\mathcal{B}'$, $\mathcal{B}''$ and $\mathcal{B}'''$.

C : chord (almost) perpendicular to D, or length of chord. See Section 3.

$\mathcal{C}$: k by k matrix. See Section 6. Similarly for $\mathcal{C}'''$.

D : diameter of R, or length of diameter. See Section 3.

F_p : finite field of p elements. See Section 6.

h : $\beta\rho/(\beta+\rho)$.

i : nonnegative integer.

j : nonnegative integer.

k : $n^{\frac{1}{2}}$. See Section 6.

K_i : constant, $1 \leq i \leq 3$. See (3.2), (3.5) and (5.5).

$\lg$: $\log_2$. $\lg^j n$ denotes $(\log_2(n))^j$.

$\ln$: $\log_e$.

L : perimeter of R, or length of the perimeter.

m : number of input bits still to be accepted. See proof of Thm. 4.1.

M : maximum number of elements of S sharing any output port. See Lemma 3.3.

n : number of bits in inputs a and b.

N : positive integer.

p : prime number ($p > 1$). ($p = nq+1$ in Section 6.)

p_i : i-th least significant output bit, *output* $= p_{2n} \ldots p_1$ in binary notation.

P_N : $\{ij \mid 0 \leq i < N, 0 \leq j < N\}$. See Section 4.

$\mathcal{P}_N$: $\{p \mid p \text{ prime}, 2 \leq p \leq N\}$.

q : positive integer.

r : M/n. See Lemma 3.3. (Free variable in Lemma 3.2.)

R : region in which computation is performed. See assumption A1.

s : number of bits of information stored in R. *See proof of Thm.* 4.1.

S : $(p_{2n-1},\ldots,p_n)$. See proof of Lemma 3.3.

S_i : subsequence of S, $1 \leq i \leq 3$.

T : time required for computation.

T_0 : constant defined by equation (5.2).

u : n-th root of unity in F_p.

$\mathcal{U}$: k by k matrix. See Section 6. Similarly for $\mathcal{U}'$.

w : k-th root of unity in F_p, $w = u^{\bar{k}}$.

$\mathcal{W}$: k by k matrix. See Section 6.

α : free variable, $0 \leq \alpha \leq 1$.

β : minimum area required to store one bit. See assumption A6.

δ : function defined by equation (4.3).

λ : minimum width of a wire. See assumption A2.

μ : $\mu(N) = |\mathcal{P}_N|$.

$\hat{\mu}$: $\hat{\mu}(N) = N^2/(0.71 + \mathrm{lglg}N)$, approximation to $\mu(N)$.

ν : maximum number of wires which can overlap at any point. See assumption A3.

ρ : minimum area for an I/O port. See assumption A5.

σ : $\sigma(N) = \sum_{p \in \mathcal{P}_{N-1}} p$.

τ : minimum time for propagation of one bit along a wire. See assumption A4.

DAT

FILM

4 –